Amendments to the Claims:

Following is a complete listing of the claims pending in the application, as amended:

1-29.   (Cancelled)

30.   (Currently Amended) A method of scheduling a plurality of components to be performed by a computing device, each component <u>having a scheduled component start time and</u> including a plurality of actions, <u>each of the plurality of actions being non-preemptible and</u> suitable for [[non-preemptive]] execution by the computing device, and each <u>of the plurality of</u> <u>non-preemptible</u> actions having a scheduled <u>action</u> start time, said method comprising:

determining an earliest <u>action</u> start time <u>from among the plurality</u> of [[the]] scheduled <u>action</u> start times <u>for the plurality of non-preemptible actions</u>;

identifying an earliest component having the earliest action with the earliest <u>action</u> start time;

executing a first action<u>, which has the earliest action start time</u> of the plurality of actions from the identified earliest component<u>,</u> to completion without preemption;

obtaining a returned event from said executed <u>first</u> action in accordance with said executing; and

propagating said returned event to a second action from dependent components of the earliest component.

31.   (Currently Amended) A method according to claim 30 wherein the plurality of components each have a unique identifier<u> used to select the earliest component from said</u> <u>plurality of components</u>, and wherein a plurality of components each have <u>an action with</u> the earliest start time, the act of identifying the earliest component comprising selecting the earliest component from the plurality of components having <u>an action with</u> the earliest start time based on the unique identifiers.

32.   (Currently Amended) A method according to claim 30 wherein the first action is selected <u>from those actions that have a same earliest start time</u> based on a predefined preference associated with the first action.

33.   (Currently Amended) A method according to claim 30, further comprising:

dividing at least one of said components into said plurality of non-preemptible actions, wherein each of said plurality of actions are scheduled suitable for non-preemptive execution and are non-preemptively executed.

34.   (Currently Amended) A method according to claim 30, further comprising:

receiving an interrupt after the act of identifying an earliest action start time and before the act of executing the first action, and in accordance therewith servicing said received interrupt and then performing again the act of identifying the earliest component having the earliest action with the earliest action start time, wherein said interrupt corresponds to one of a shell command and a connection request.

35.   (Previously Presented) A method according to claim 30, further comprising:

receiving an interrupt after the act of identifying an earliest component and before the act of executing, and in accordance therewith identifying a component corresponding to said interrupt and a reactive action therein, and in accordance therewith executing said reactive action instead of the identified earliest action, wherein said interrupt corresponds to an alert input.

36.   (Currently Amended) A method according to claim 30, further comprising:

receiving an interrupt after the act of identifying an earliest component and before the act of executing, and in accordance therewith servicing said interrupt and then identifying a component corresponding to said interrupt and a reactive action therein, and in accordance therewith executing said reactive action instead of the identified first action, wherein said interrupt corresponds to one of a hardware interrupt having an interrupt service routine and a software interrupt having said interrupt service routine.

37.   (Previously Presented) A method according to claim 30, wherein each component of said plurality of components further includes a plurality of states, and the method further comprising: updating one or more state of said plurality of states related to the earliest component, after said propagating.

38.   (Currently Amended) A computer-readable medium encoded with a computer program code for scheduling a plurality of components to be performed by a computing device, each component having a scheduled component start time[[, and each component]] and including a plurality of actions, each of the plurality of actions being non-preemptible and suitable for [[non-preemptive]] execution by the computing device, and each of the plurality of non-preemptible actions having a scheduled action start time, the program code causing the computing device to execute a method comprising:

determining an earliest action start time from among the plurality of [[the]] scheduled action start times for the plurality of non-preemptible actions;

identifying an earliest component having the earliest action start time;

executing a first action, which has the earliest action start time of the plurality of actions from the identified earliest component to completion without preemption;

obtaining a returned event from said executed first action in accordance with said executing; and

propagating said returned event to a second action from the earliest component.

39.   (Currently Amended) A computer program product according to claim 38 wherein the plurality of components each have a unique identifier, and wherein a plurality of components each have an action with the earliest start time, the act of identifying the earliest component comprising selecting the earliest component from the plurality of components having the earliest start time based on the unique identifiers.

40.   (Previously Presented) A computer program product according to claim 38 wherein the first action is selected based on a predefined preference associated with the first action.

41.   (Previously Presented) A computer program product according to claim 38, the method further comprising:

dividing at least one of said components into said plurality of actions, wherein each of said plurality of actions are suitable for non-preemptive execution.

42.   (Currently Amended) A computer program product according to claim 38, the method further comprising:

receiving an interrupt after the act of identifying an earliest action start time and before the act of executing the first action, and in accordance therewith servicing said received

interrupt and then performing again the act of identifying the earliest <u>component having the earliest action with the earliest action</u> start time, wherein said interrupt corresponds to one of a shell command and a connection request.

43.   (Previously Presented) A computer program product according to claim 38, the method further comprising:

receiving an interrupt after the act of identifying an earliest component and before the act of executing, and in accordance therewith identifying a component corresponding to said interrupt and a reactive action therein, and in accordance therewith executing said reactive action instead of the identified earliest action, wherein said interrupt corresponds to an alert input.

44.   (Previously Presented) A computer program product according to claim 38, the method further comprising:

receiving an interrupt after the act of identifying an earliest component and before the act of executing, and in accordance therewith serving said interrupt and then identifying a component corresponding to said interrupt and a reactive action therein, and in accordance therewith executing said reactive action instead of the identified first action, wherein said interrupt corresponds to one of a hardware interrupt having an interrupt service routine and a software interrupt having said interrupt service routine.

45.   (Previously Presented) A computer program product according to claim 38, wherein each component of said plurality of components further includes a plurality of states, and the method further comprising:

updating one or more state of said plurality of states related to the earliest component, after said propagating.

46.   (Currently Amended) A system for scheduling a plurality of components, the system comprising:

a computing device configured to receive a plurality of components, each component having a scheduled <u>component</u> start time and each component including a plurality of actions<u>, each of the plurality of actions being non-preemptible and</u> suitable for non-preemptive execution by the computing device<u>, and each of the plurality of non-preemptible actions having a scheduled action start time</u>; and

a scheduling program in communication with the computing device, the scheduling program configured to:

determine an earliest action start time from among the plurality of [[the]] scheduled action start times for the plurality of non-preemptible actions;

identify an earliest component having the earliest action start time;

select for execution by the computing device, to completion without preemption, a first action which has the earliest action start time of the plurality of actions from the identified earliest component;

obtain a returned event from said executed first action; and

propagate said returned event to a second action from the earliest component.

47.   (Currently Amended)   A system according to claim 46 wherein the plurality of components each have a unique identifier, and wherein a plurality of components each have an action with the earliest start time, the act of identifying the earliest component comprising selecting the earliest component from the plurality of components having the earliest start time based on the unique identifiers.

48.   (Previously Presented) A system according to claim 46 wherein the scheduling program selects the first action based on a predefined preference associated with the first action.

49.   (Previously Presented)   A system according to claim 46, wherein the scheduling program is further configured to:

divide at least one of said components into said plurality of actions, wherein each of said plurality of actions are suitable for non-preemptive execution.

50.   (Currently Amended) A system according to claim 46, wherein the scheduling program is configured to receive an interrupt, and if the interrupt is received after the identification of the earliest action start time and before the selection for execution, the scheduling program is configured to select the interrupt for execution and then again identify the earliest component having the earliest action with the earliest action start time, wherein said interrupt corresponds to one of a shell command and a connection request.

51.   (Previously Presented)   A system according to claim 46, wherein each component of said plurality of components further includes a plurality of states, and computing device further configured to:

update one or more state of said plurality of states related to the earliest component, after said propagating.

Add Claims 52-63 as follows:

52.   (New) A method according to claim 30, wherein all of the components to be performed by the computing device are performed non-preemptively using non-preemptible actions implemented as function calls without context switching.

53.   (New) A computer-readable medium encoded with a computer program code as in claim 38, wherein all of the components to be performed by the computing device are performed non-preemptively using non-preemptible actions implemented as function calls without context switching.

54.   (New) A system according to claim 46, wherein all of the components to be performed by the computing device are performed non-preemptively using non-preemptible actions implemented as function calls without context switching.

55.   (New) A method according to claim 30, wherein the determining an earliest action start time is a dynamic determination wherein the earliest action start time is computed dynamically at the end of execution of each action.

56.   (New) A method according to claim 30, wherein each action is executed as a function call, and each action includes one or more instructions.

57.   (New) A method according to claim 30, wherein there is no priority-based scheduling of actions, and no preemption of any executing actions, including no preemption of an executing action by an interrupt.

58.   (New) A method according to claim 30, wherein the method provides a scheduler that examines the actions which are implemented as function calls, and schedules them so that all

of the ready actions are executed in the order they are required to be executed by the application.

59.   (New) A method according to claim 30 wherein:
        the plurality of actions having at least one relationship there between;

        the plurality of components each have a unique identifier used to select the earliest component from said plurality of components, and wherein a plurality of components each have an action with the earliest start time, the act of identifying the earliest component comprising selecting the earliest component from the plurality of components having an action with the earliest start time based on the unique identifiers;

        the first action is selected from those actions that have a same earliest start time based on a predefined preference associated with the first action;

        each component of said plurality of components further includes a plurality of states, and the method further comprising: updating one or more state of said plurality of states related to the earliest component, after said propagating; and

        receiving an interrupt and in response to receiving the interrupt performing one of:

        (i) when the interrupt is received after the act of identifying an earliest action start time and before the act of executing the first action, and in accordance therewith servicing said received interrupt and then performing again the act of identifying the earliest component having the earliest action with the earliest action start time, wherein said interrupt corresponds to one of a shell command and a connection request;

        (ii) when the interrupt is received after the act of identifying an earliest component and before the act of executing, and in accordance therewith identifying a component corresponding to said interrupt and a reactive action therein, and in accordance therewith executing said reactive action instead of the identified earliest action, wherein said interrupt corresponds to an alert input; and

        (iii) when the interrupt is received after the act of identifying an earliest component and before the act of executing, and in accordance therewith servicing said interrupt and then identifying a component corresponding to said interrupt and a reactive action therein, and in accordance therewith executing said reactive action instead of the identified first action, wherein said interrupt corresponds to one of a hardware interrupt having an interrupt service routine and a software interrupt having said interrupt service routine.

60.   (New) A method according to claim 59, wherein each of the plurality of components consists of a plurality of actions.

61.   (New) A method according to claim 30, wherein each of the plurality of components consists of a plurality of actions.

62.   (New) A computer program product according to claim 38 , wherein each of the plurality of components consists of a plurality of actions.

63.   (New) A system according to claim 46, wherein each of the plurality of components consists of a plurality of actions.